Soundsquatting: Uncovering the use of homophones in domain squatting

Nick Nikiforakis[1], Marco Balduzzi[2], Lieven Desmet[3], Frank Piessens[3], and Wouter Joosen[3]

[1] Department of Computer Science, Stony Brook University
nick.nikiforakis@cs.stonybrook.edu
[2] TrendMicro
marco.balduzzi@iseclab.org
[3] iMinds-DistriNet, KU Leuven, 3001 Leuven, Belgium
{firstname.lastname@cs.kuleuven.be}

Abstract In this paper we present *soundsquatting*, a previously unreported type of domain squatting which we uncovered during analysis of cybersquatting domains. In soundsquatting, an attacker takes advantage of homophones, i.e., words that sound alike, and registers homophone-including variants of popular domain names. We explain why soundsquatting is different from existing domain-squatting attacks, and describe a tool for the automatic generation of soundsquatting domains. Using our tool, we discover that attackers are already aware of the principles of soundsquatting and are monetizing them in various unethical and illegal ways. In addition, we register our own soundsquatting domains and study the population of users who reach our monitors, recording a monthly average of more than 1,700 non-bot page requests. Lastly, we show how sound-dependent users are particularly vulnerable to soundsquatting through the abuse of text-to-speech software.

1 Introduction

Due to its critical position, DNS has, over the years, attracted many attacks targeting various parts of the protocol and the DNS infrastructure. These attacks can be grouped in the following targeted categories: protocol weaknesses (e.g., DNS cache poisoning [14,25]), vulnerable implementations of DNS servers (e.g., buffer overflows in BIND [20]), and the interaction of users with DNS. Of all the aforementioned categories, we postulate that the attacks targeting the user-DNS interaction are the hardest to eliminate, since they involve the education of the entire current and future Internet population, rather than the technical correction of a protocol shortcoming, or a software vulnerability.

One of the ways users interact with DNS is through the typing of domain names in their browsers' address bar. Attackers realized early on that users make spelling mistakes when typing the domain name of their desired destinations and started registering these "typo-including" domains in order to capitalize on the potential incoming traffic. This practice was named *typosquatting* [19,27],

and typosquatters use these domains in a wide range of unethical and illegal ways including showing paid ads of competitors [21], and the exfiltration of user credentials through phishing [10]. In addition to typosquatting, other variations of domain squatting have been proposed over time, such as homograph attacks [11,16] where the attacker abuses the visual similarity of two characters from different character sets to construct domains that have the appearance of a popular authoritative domain but lead to different destinations.

In this paper we present *soundsquatting*, a domain squatting technique which we uncovered while researching generic cybersquatting. Soundsquatting takes advantage of the sound similarity of words and the user's confusion of which word represents the desired concept. The attack is based on *homophones*, i.e., sets of words that are pronounced the same but are spelled differently, e.g., {*ate, eight*}. Soundsquatting is different from typosquatting in that it does not rely on typing mistakes and in that not all domains contain homophones and thus, not all domains can be soundsquatted.

To evaluate soundsquatting, we compile an English homophone database and we design *AutoSS* (AutoSoundSquatter), a tool which, given a list of target domains, generates valid soundsquatting domains. For the Alexa top 10,000 websites, AutoSS was able to generate 8,476 soundsquatting domains, 1,823 (21.5%) of which were already registered. Through a series of automatic and manual experiments, we categorize these registered domains and discover that, even though homophone-based domain squatting has not appeared in literature around cybersquatting, its principles are known and practiced by cybersquatters, albeit to a lesser extent than typosquatting. Using data that we obtain through crawling, we show that soundsquatting is being used for displaying ads on parked domains, stealing traffic from target domains, performing affiliate scams, conducting phishing attacks and installing malicious software on unsuspecting visitors.

In addition to studying the use of already registered soundsquatting domains, we register 30 available ones and study the population of users that reached our domains, recording a monthly average of 1,718 requests from real users, originating from 123 countries, showing that users are indeed susceptible to homophone confusion. Finally, we examine six popular software screen readers and show how they can all be abused to perform soundsquatting attacks against sound-dependent users who rely on text-to-speech software.

Overall, our findings show that soundsquatting can be abused in exactly the same way as typosquatting, and thus should be taken into account by owners of large websites when they are trying to protect their brand-names and customers.

Our main contributions are:

- We uncover a, previously unreported, domain-squatting attack, based on homophone-confusion rather than typographic mistakes, which we name *soundsquatting*, and present the architecture of a tool capable of automatically generating soundsquatting domains.
- We perform a systematic, large-scale analysis of the existing soundsquatting domains targeting the Alexa top 10,000 sites and highlight their abuse.

– We actively measure the worldwide population of users who make homophone mistakes, confirming the validity and practicality of the attack.
– We show how soundsquatting can be used against sound-dependent users.

2 Soundsquatting

In this section, we introduce all the necessary terminology for soundsquatting and describe in detail the workings of AutoSS, our tool for the automatic generation of soundsquatting domains. Lastly, we examine the soundsquatting domains that our tool generated, when targeting the Alexa top 10,000 sites.

2.1 Terminology

Homophones are sets of words that have the same pronunciation. Homophones can be spelled differently but have the same meaning, such as {*guarantee, guaranty*} or spelled differently and have a different meaning, such as {*whether, weather*} and {*idle, idol, idyll*}.

Given the aforementioned definition of homophones, we define *soundsquatting* as the practice of registering domain names that contain words that are homophones of authoritative domains and *soundsquatters* the individuals or organizations involved in the activity of soundsquatting. As in generic domain squatting, *authoritative* domains are domains that are targeted by soundsquatters, and usually belong to high-traffic websites with millions of visitors. The more legitimate users a website has, the more users are likely to land on the website connected to the soundsquatting domain. When an authoritative domain is targeted by a soundsquatting attack, this domain is called *soundsquatted*.

For instance, assuming `weatherportal.com`, an authoritative weather site, a soundsquatter can register the domain `whetherportal.com`, in order to capture the traffic of users who mistakenly type the word "whether" instead of "weather". When users type the wrong word and reach the soundsquatting domain, the soundsquatter, like generic domain squatters, can then monetize their visit in a wide range of unethical and illegal ways.

2.2 Differences with Typosquatting

Before moving on to the discovery and study of soundsquatting domains, it is important to differentiate *soundsquatting* from *typosquatting*. As the term reveals, typosquatting involves "typos", i.e., misspellings of domain names, usually associated with typing mistakes. In 2006, Wang et al. categorized the typos involved in typosquatting in five different categories [27]. Assuming the domain `example.com` and the intended URL `www.example.com`, the five proposed categories are the following:

1. **Missing-dot typos**: The dot following "www" is forgotten, e.g., `wwwexample.com`

2. **Character-omission typos**: One character is omitted, e.g., `www.exmple.com`
3. **Character-permutation typos:** Consecutive characters are swapped, e.g., `www.examlpe.com`
4. **Character-replacement typos:** Characters are replaced by their adjacent ones, given a specific keyboard layout, e.g., `www.ezample.com` where "x" was replaced by the QWERTY-adjacent "z".
5. **Character-insertion typos:** Characters are mistakenly typed twice, e.g., `www.exaample.com`

Later research in typosquatting showed that in addition to the above classes of typos, domain squatters are also registering authoritative domains under different, less-popular top-level domains [4].

In all of the above cases, users *intend* to type a specific URL, but accidentally mistype the URL and initiate a request for the wrong page, before realizing their mistake. Contrastingly, in soundsquatting, users type exactly what they were planning to, but their intended destination is a different one. The mistake happens at a word-level, rather than a character-level, and the substituted words are real dictionary words and not mistypes of other words. The confusion between the intended word and the typed one, is further amplified when a domain contains a homophone belonging to a set of homophones with the same meaning. Consider `guarantybanking.com`, a domain belonging to a banking website. As mentioned earlier, "guarantee" is a homophone of "guaranty", and `guaranteebanking.com` is, at the time of this writing, parked and available for sale. In such cases, the typing of the "correct" domain, involves the memorization of one specific spelling, rather than a translation from concepts into words. It is also difficult to predict, if a person hears about "Guarantee Banking" for the first time, which spelling will she choose to use.

2.3 Generating soundsquatting domains

Any system that is geared towards the discovery of domain-squatting activity requires at least the following two resources: a set of target, authoritative domains and a list of rules and models for the transformation of authoritative domains and the generation of possible squatting domains. In the case of typosquatting, these rules may be the neighboring characters of every key under a specific keyboard layout and models of character omission, duplication and replacement. For soundsquatting, these resources are the following:

Authoritative domain list: Under the assumption that popular domains are targeted more than less popular domains, we obtained a list of the top 10,000 Internet websites, according to Alexa. In Section 2.4 we provide the number of unique domains contained in this list.

Dictionary: A dictionary (or wordlist) is required for the extraction of valid words from domain names. For instance, given a sufficiently large dictionary and the domain `youtube.com`, an algorithm can straightforwardly search for

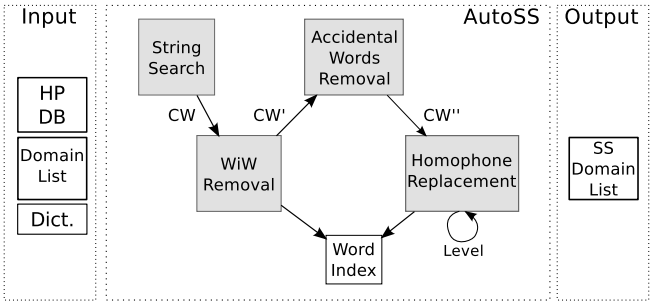

Figure 1. The architecture of AutoSS. Given a homophone database, a list of target domains and a dictionary, AutoSS outputs a list of possible soundsquatting domains

the presence of all words in that domain (excluding the top-level domain) and conclude that the domain is comprised out of the words "you" and "tube".

Transformation rules: Apart from a dictionary, a database of English homophones is also required. We compiled a homophone database, by scraping `homophone.com`, a website dedicated to homophones, and Wikipedia's list of dialect-independent homophones [28]. In addition, we manually added to our homophone database the list of numbers from one to one hundred together with their appropriate word form, e.g., {*9, nine*}, and a few common idioms used regularly in Internet slang, e.g., {*you, u*}.

To automatically generate soundsquatting domains, we created AutoSS (AutoSoundSquatter), a tool which receives as input the aforementioned resources and generates valid soundsquatting domains – see Figure 1. AutoSS operates as follows. After the loading to memory of the homophone database and the dictionary, each entry in the Alexa list of websites is parsed, in order to isolate the main domain, from its domain extension and possible subdomains and paths. If the resulting string contains dashes (-), then we perceive this as a sign, from the domain owner, of separation of words; e.g., `search-results.com` can be separated to the words "search" and "results" without the aid of a dictionary. If the domain does not contain dashes, then we perform a string search for the presence of every word in our dictionary, in the domain. While this is a relatively fast process, the resulting set of candidate words (CW in Figure 1) requires substantial processing, mainly due to candidate words included in other candidate words and the presence of accidental words. Below, we describe these issues and our techniques for automatically detecting and resolving them:

Words-in-Words Removal: Consider the domain `linkedin.com` and the homophone set {*in, inn*}. Even though we would, ideally, want to discover just the words "linked" and "in", a typical dictionary search will discover the words: {*in, ink, inked, ked, link, linked*}. The obvious next step would be to delete all

candidate words words that are contained in others. The issue, however, is that, while the words {*in, ink, inked, ked, link*} are all contained in the word "linked", a removal of the word "in" from the candidate words is wrong (since the word exists on its own accord after the word "linked") and by doing so we miss the opportunity of generating the soundsquatting domain `linkedinn.com`. In our implementation we solve this problem as follows:

Whenever a pair of words {*a, b*} is found, where *a* is included in *b*, *b* is replaced, in the domain name, by another string of equal length. After this replacement, the domain name is searched again for the presence of word *a*. If the word is still found, then *a* is not deleted from the set of candidate words. Thus, in our earlier example and the pair of words {*in, linked*}, `linkedin.com` is transformed to `_____in.com`. Since the word "in" is still found in the domain name, it is not removed from the candidate words. Before proceeding, our tool also records the index of the word's location in the transformed domain in the "Word Index" component, so that when, at a later step, words are replaced by their homophones, our tool replaces the appropriate "in", avoiding results such as `linnkedinn.com`, which do not conform to our definition of soundsquatting since "linnked" is neither a valid dictionary word, nor a homophone of any other word. At the end of this process, our candidate words set is reduced to {*linked, in*} (CW′ in Figure 1), which is the desirable outcome.

Accidental Words Removal: This module receives the, possibly modified, candidate-words set from the Words-in-Words module and attempts to identify and remove accidental words from the candidate words. The issue of accidental words can be illustrated as follows. Consider the domain `leaseweb.com`, belonging to a web-hosting provider. The ideal word breakdown would be {*lease, web*}. Given our dictionary and the previous step of selective removal of words that are included in others, AutoSS would discover the words {*lease, sew, web*}. In this set, the word "sew" is included since it is a dictionary word, which accidentally appears in the domain name, formed by the last two letters of the word "lease" and the first letter of the word "web". We partially solve this problem by attempting to exhaustively create permutations of the candidate words, e.g.,:

```
lease       ✗
...
websew      ✗
...
leasesew    ✗
leaseweb    ✓
```

until either the permutation perfectly matches the target domain name (CW″), or, due to the exponential nature of permutations, the computation reaches a timeout. If the timeout is reached before the module finishes, then AutoSS falls back to the candidate word list that was received by the WiW Removal.

Homophone Replacement: In this last module, AutoSS uses the set of candidate words discovered by the previous modules and generates new domains, through the replacement of one homophone word by another. To this end, the module queries its homophones database for each candidate word, and for each

homophone discovered the system generates a new soundsquatting domain, by replacing the candidate word with a homophone. For every candidate word, the module takes into account information found in the Word Index, so as to replace the right word in the aforementioned corner cases.

In addition to single replacements of homophones, AutoSS has a "Level" parameter (as shown in Figure 1), which specifies the number of concurrent homophone replacements for domain names with more than one homophones discovered. Consider the case of `thepiratebay.se`, a popular Torrent tracker. AutoSS will discover the homophones {*the, thee*} and {*bay, bey*}. While these can be used to create the soundsquatting domains `theepiratebay.se` and `thepiratebey.se`, a third domain can be generated by replacing both at the same time, i.e., `theepiratebey.se`. For our experiments we chose a Level of two, in order to limit AutoSS to maximum two homophone replacements at a time, even if a domain contains multiple homophones. While a higher Level would allow significantly more combinations and thus generate many more soundsquatting domains, we reasoned that three or more homophone mistakes in a single domain name are unlikely to occur, and thus decided against it.

AutoSS Limitations: Due to the flexibility of the English language and the freedom of infinite word-plays, our system's techniques for isolating words in domain names are necessarily heuristic-based. In Section 6, we estimate the amount of false positives generated by AutoSS, and briefly discuss possible ways of further lowering false positives that could be pursued in future work.

2.4 Results

From the Alexa list of top 10,000 Internet websites, we extracted 9,926 *Public Suffix + 1* domains. Given these domains and our homophone database which contains 2,913 words belonging to 1,337 sets of homophones, AutoSS extracted a total of 6,418 homophones from the list of domains which, when combined up to a level of two, generated 8,476 soundsquatting domains. Interestingly, in our results we saw that 67.3% of the domains contained no homophones.

The highest ranking domain which contained homophones was `youtube.com`, for which AutoSS generated the domains: `yewtube.com`, `ewetube.com` and `utube.com`. The domain with the most homophones was `wearehairy.com`, with a ranking of 5,663 in the Alexa list of websites, containing 12 different homophones which, when combined, allowed for 32 different soundsquatting domains. From the 1,337 sets of homophones, 568 (42.48%) were used at least once to generate a soundsquatting domain. Table 1 shows the ten homophone sets used the most by AutoSS over the Alexa list of websites.

In Figure 2, we explore the correlation between the ranking of a website and the number of homophones found in its domain name. The scatterplot reveals that there is no significant relationship between the two, meaning that, on average, high ranking websites are as vulnerable to soundsquatting as low-ranking ones. This is an experimental validation of what we intuitively expected: homophones are dependent on the choice of words, which is unrelated to the popularity of a website, at least within the top Alexa sites.

Homophone set	#Times Utilized
{*2, two, to, too*}	735
{*1, one, won*}	300
{*ere, air, aire, are, ayr, ayre, err, eyre, heir*}	278
{*four, 4, for, fore*}	250
{*bi, buy, by, bye*}	223
{*do, dew, due, doe, dough*}	208
{*whirled, whorled, world*}	156
{*yew, you, ewe, u*}	150
{*cite, sight, site*}	134
{*0, zero, -xero*}	134

Table 1. Top ten homophone sets for word replacements in the Alexa top 10K sites

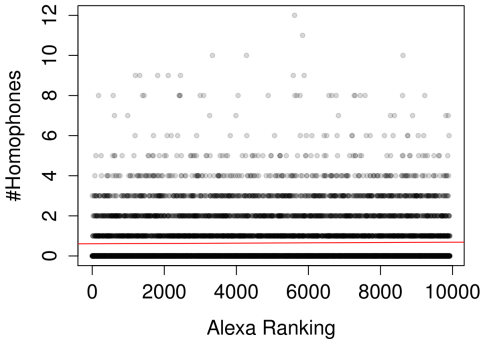

Figure 2. Scatterplot showing the lack of a significant correlation between a website's popularity and the number of homophones contained in its domain name (r=0.019)

3 Evaluation of Soundsquatting

In this section, through a series of automated and manual experiments, we analyze the existing, i.e., already registered, soundsquatting domains and categorize them according to the purpose that they serve.

3.1 Method of Categorization

In Section 2.3 we described AutoSS, our system for the automatic generation of soundsquatting domains, and we showed that it was able to generate 8,476 soundsquatting domains targeting the Alexa top 10,000 sites. In order to find out whether domain squatters are already aware of homophones and the principles of soundsquatting, we performed a two-step process to identify the domains that were already registered. First, we tried to resolve all domains to their IP addresses. A domain that successfully resolves is obviously registered, however a domain that does not resolve may still be registered but not assigned a valid IP address. Thus, for the set of domains that did not resolve to an IP address, we performed WHOIS lookups, and tried to register them with a popular domain name registrar. At the end of this process, we identified that 1,823 domains (21.5% of the total domains generated) were already registered.

In order to classify the registered domains, we implemented a crawler based on *PhantomJS* [15] that visited each of the 1,823 domains, waited for ten seconds (to allow the loading of remote content) and then took a screenshot of the page as well as recorded the HTML and final URL for later processing. The final URL was used in order to detect redirections from the visited soundsquatting domain to a different domain. To categorize each site, we followed a semi-automatic approach. We begun by manually skimming over the screenshots of all pages and

grouping together images that looked alike. The majority of these were *parked pages*, i.e., pages that show ads, somewhat relevant to the domain name and usually also advertise that the domain may be for sale. Other groups were pages with little content, stating that the site is "under construction", placeholder pages by popular registrars informing their clients how to setup a website on their registered domain, and pages containing generic errors, such as `404 Forbidden`. For each group, we examined the corresponding HTML of a few domains and created generic HTML- and JavaScript-signatures that could automatically categorize the remaining pages within each group. Using this approach, our page-characterizing scripts could eventually automatically classify 77.2% of all the crawled domains. The remaining unclassified domains (417) were classified manually by visiting each website and carefully inspecting the source code, available WHOIS information and any similarities (visual-, content- and audience-based) between the soundsquatting domain and the corresponding authoritative one.

3.2 Categorization results

By combining the results of the automatic classification and those of our manual investigation, we categorized the registered soundsquatting domains as follows:

Authoritative-owned domains: From the 1,823 studied domains, we identified 155 soundsquatting domains which belonged to the companies and organizations behind the corresponding authoritative domains. In the vast majority of cases, the user would be automatically redirected to the correct authoritative domain without warnings or additional dialogues. The redirect almost always happened through a 301/302 HTTP status code and occasionally users were redirected to one or two intermediate hosts, which would in-turn redirect them to the appropriate domain. In these cases, we were able to identify that the intermediate hosts belong to brand-protecting companies which were, most likely, registering the fact that a user did a specific mistake, before redirecting her to the appropriate destination.

In two instances, the owners of the authoritative domains were attempting to educate their users about homophone confusion. For instance, `myfreepaysight.com` is a soundsquatting domain for the adult site `myfreepaysite.com`. When the soundsquatting domain is visited, users greeted with a message that points out the difference between the two domains.

Parked/Ads/For Sale domains: Parked domains have been identified by prior research as the preferred monetizing way of domain squatters [21,27]. As we mentioned earlier, these domains contain no real content, except ads which are constructed on demand, usually by a domain-parking agency, based on the words included in a domain name and preferences by the owner of the domain. In this category, we also include domains that were found to show ads without being affiliated with a large domain-parking agency, e.g., `net0.net`, a soundsquatting domain targeting `netzero.net`, as well as domains that are listed as "for-sale". In total, ad-driven domains represent the largest chunk of existing soundsquatting domains, with 954 cases (52.3%).

Affiliate-abusing domains: While examining the soundsquatting domains that redirected the user to the appropriate authoritative domain, we realized that 32 soundsquatting domains were abusing affiliate programs of the corresponding authoritative domains. In affiliate programs, existing customers of a website are encouraged to bring new customers, through the promise of a small commission for every brought customer.

In affiliate abuse, an attacker takes advantage of an affiliate program of a website by appending his own affiliate identifier to unsuspecting visitors. More specifically, consider the domain `mybrowsercache.com` which is a soundsquatting domain for `mybrowsercash.com`. At the time of this writing, when a user visits the former domain, she will automatically be redirected to `http://www.mybrowsercash.com/index.php?refid=312044`. Notice that a specific referrer identifier is added to the URL. In this way, the attacker who registered `mybrowsercache.com` will gain a commission every time that a user confuses "cash" and "cache", and subsequently registers to the target website. In addition to soundsquatting domains redirecting to sites with affiliate programs, we also recorded some cases where the user was kept on the soundsquatting domain and the authoritative site, together with the attacker's affiliate identifier, was opened in a full-page HTML frame.

Hit-stealing domains: From our analysis, we discovered 22 cases where attackers used soundsquatting to capture traffic and feed their own "business-related" domains with hits intended for the authoritative targeted site. In fact, in the majority of cases, both the soundsquatting and corresponding authoritative domain had content belonging to the same category, but they were owned by different organizations and individuals. From our experiments, we found that the most targeted business categories are adult, online shopping and travel. Below, is a list of a few examples:

- `ashemailtube.com` is a soundsquatting domain for the domain `ashemaletube.com`, a transvestite-oriented porn site. When the soundsquatting domain is visited, the user is redirected to `trannydates.com` a dating site specifically for transvestites.
- `video-1.com`, a soundsquatting domain for the adult video portal `video-one.com`, currently hosts an online sex shop
- `todomains.ru` is selling domain-registration services and is a soundsquatting domain for `2domains.ru`, a large Russian domain registrar.
- `gamefive.com` is a soundsquatting domain for `game5.com`, an online gaming site. The soundsquatting domain had been for sale for three years before being turned into an online gaming site.
- `textsail.ru` is a soundsquatting site for `textsale.ru`. Both sites sell articles and stories on a wide range of topics.

In this category, we also included soundsquatting domains that profit from the trustworthiness associated with well-known and popular authoritative domains, in order to advertise non-related websites. In these cases, it is not necessary that the category of the soundsquatting domain matches the category of

the targeted authoritative domain. For example, the owner of `freemale.hu` is probably exploiting the popularity of the well-known Hungarian e-mail provider `freemail.hu` to advertise his own web page, in the same way that the soundsquatting domain `tvto.no` is abusing the popularity of the Norwegian TV2 channel `tv2.no` and subsequently redirects the user to an online casino.

Scam-related domains: Soundsquatting domains can also be used for scamming purposes. We identified 16 cases of domains used to perform different form of scams, generally by luring visitors into subscribing to fake lotteries and surveys. For instance, `vhone.com`, a soundsquatting domain targeting `vh1.com`, redirects the user to a survey site, where users are promised an opportunity to win high-end electronics in return to their participation in a short survey. The users are then trapped in a series of redirects where they are constantly promised more and more prizes while they divulge more and more private information, such as their names, email addresses and mobile phone numbers.

Promoting-related domains: In this category, we included seven domains that were used to promote someone or something related to the authoritative domain. For example, `teambeechbody.com` is a soundsquatting domain for `teambeachbody.com`, an online fitness club where people can subscribe as "fitness coaches" and gain a commission for every user that they successfully coach. At the time of this writing, when the soundsquatting domain is visited, the user is redirected to the page of a specific coach within the authoritative `teambeachbody.com` domain, giving that specific coach higher chances of getting selected to coach a user, over other coaches on the website. In another case, the authoritative `readnovel.com` domain is soundsquatted by `rednovel.com`, which redirects the user to `http://www.lvse.com/site/readnovel-com-3550.html`. The page contains a review of the authoritative website, providing a safety score, user comments and similar websites.

Others domains: We conclude our analysis with six soundsquatting domains used for malicious purposes, e.g., to install malware and acquire private information. Movreel (`movreel.com`), is a free-of-charge service for streaming movies, and `movreal.com` is one of its soundsquatting domains. At a first glance, `movreal.com` appears to be another streaming provider for movies, but interestingly the user is requested to download a browser plug-in (`AVS_Media_Player.exe`) to watch the video. The offered plug-in, however, is malicious and detected by most antivirus vendors as a variant of the `solimba` malware, i.e., an installer for other malicious software and a provider of adware campaigns. Similarly, `utube.com`, a soundsquatting domain for `youtube.com`, makes use of videos to social-engineer the user into first divulging her personal information and then, depending on the type of browser used, installing a browser extension. The extension installed when using Mozilla Firefox, injects unwanted search results, launches pop-ups, and gathers user statistics.

Among the other cases, we identified two domains that are likely used to acquire private user information, in particular email credentials. One of these is `innbox.lv`, which is the soundsquatting domain of the well-known Latvian

service provider *InBox*, where both sites offer free e-mail accounts. Finally, we were able to confirm two soundsquatting domains involved in phishing campaigns against e-commerce and business-related sites.

Summary: Overall, by combining the results of the outlined categories, out of the 1,823 registered soundsquatting domains, 1,037 (56.88%) were categorized as malicious. For the remaining domains, 155 of them belong to the corresponding authoritative domains' owners, 300 domains are registered under different organizations that are using them in a legitimate way, and 331 domains were offline, or showing HTTP errors, or under construction when we visited them.

4 User Characterization

In previous sections, we categorized the registered soundsquatting domains according to their use. We now turn our attention to the users who, due to homophone confusion, land on soundsquatting domains and study their population.

As described in Section 2.4, AutoSS was able to generate 8,476 soundsquatting domains for the Alexa top 10,000 websites. From these, 1,823 domains (21.5%) were already registered, leaving 6,653 unregistered soundsquatting domains. To actively measure the worldwide population of users, and to assess the viability of the soundsquatting attack, we decided to register our own soundsquatting domains and monitor the requests towards them. Since there is no prior research on soundsquatting, there was no objective or historical way of assessing which of the unregistered domains would attract more users than others. For this reason, we manually examined the list of available soundsquatting domains from which we selected a total of 30 domains, trying to cover a wide range of soundsquatting errors.

The first three columns of Table 2 show 20 of the 30 authoritative domains targeted, the pair(s) of used homophones and the registered soundsquatting domains.While three of targeted domains could be associated with typosquatting, e.g., `theefreedictionary.com`, the rest are radically different from domains which researchers have, over the years, associated with typosquatting, e.g., `prophetclicking.com`. Most domains were registered in December 2012 while some additional ones were registered in March 2013. To present a uniform view of traffic, we provide the monthly average number of requests that each domain received, till December 11, 2013.

We resolved all domains, subdomains and requests for specific file paths to a single blank page, while recording each request's details in a set of Apache log files. Users were not automatically redirected to the appropriate authoritative domain, to avoid reinforcing the behavior of typing the wrong domain, but rather to make users aware of their mistake. We discuss our ethical considerations regarding this experiment, in the Appendix of this paper.

The last column of Table 2 shows the monthly average number of human requests received during the monitored period, and the percentage of human requests over all requests. To assess the impact of soundsquatting on humans, we needed to separate visiting bots from visiting human users. To the best of our

Auth. Domain	Homophone pair	SS Domain	#Human Req. (per month)	
thefreedictionary.com	{*the,thee*}	theefreedictionary.com	283	(39.86%)
fc2.com	{*2, too*}	fctoo.com	165	(44.84%)
jimdo.com	{*do, doe*}	jimdoe.com	150	(38.27%)
turbobit.net	{*bit, bitt*}	turbobitt.net	132	(36.07%)
leboncoin.fr	{*coin, quoin*}	lebonquoin.fr	110	(74.32%)
adserverplus.com	{*ad, add*}	addserverplus.com	98	(60.49%)
profitclicking.com	{*profit, prophet*}	prophetclicking.com	56	(48.28%)
hostgator.com	{*gator, gaiter*}	hostgaiter.com	45	(45.92%)
sitesell.com	{*sell, cel*}	sitecel.com	44	(40.00%)
discuz.net	{*disc, disk*}	diskuz.net	43	(40.19%)
tube8.com	{*8, ait*}	tubeait.com	42	(43.30%)
clixsense.com	{*sense, scents*}	clixscents.com	40	(44.44%)
a8.net	{*8, eight*}	aeight.net	48	(43.24%)
newegg.com	{*new, gnu*}	gnuegg.com	37	(36.63%)
redtubelive.com	{*red, read*}	readtubelive.com	44	(51.76%)
fiverr.com	{*err, air*}	fivair.com	33	(37.93%)
exoclick.com	{*click, clique*}	exoclique.com	32	(45.71%)
theglobeandmail.com	{*mail, male*}	theglobeandmale.com	35	(38.46%)
pastebin.com	{*bin, been*}	pastebeen.com	35	(39.77%)
ku6.com	{*6, sics*}	kusics.com	28	(33.33%)
...	...	...	...	
Total Requests per Month (30 domains):			1,718	

Table 2. A list of 20 out of the 30 registered soundsquatting domains, the corresponding authoritative domains, the homophone pair used, the monthly average number of human requests received, as well as the percentage human traffic against total traffic.

knowledge, there is no single, generic technique that can perfectly separate bots from humans. If such a technique would exist, attackers would use it to perfectly evade malware researchers, by detecting *all* high-interaction honeypots and never presenting them with malicious code.

For our purposes, we separated the requests as follows: during preliminary manual inspection of the recorded requests, we noted which requests had non-standard user-agents. Using keywords extracted from these requests, we assembled a set of nine generic identifiers, like "spider", "bot" and "crawl", that many bots have in common. In addition to these generic identifiers, we scraped 707 specific bot signatures from `useragentstring.com`. As a result, if the user agent contained any of the 716 bot signatures in our set, the request was classified as belonging to a bot. To account for bots which do not identify themselves, we also queried the IP address of each request, in the blacklist provided by `stopforumspam.com`, i.e., a database containing hundreds of thousands of IP addresses, belonging to known forum spamming bots. Lastly, each address was queried in a list of IP addresses used by well-known search engine spiders [1].

Table 2 shows that our set of 30 soundsquatting domains received an average of 1,718 human requests per month. The monthly average of total requests (not shown in the table) was 4,150. The domain that received the most hits, `theefreedictionary.com`, is a domain that can also be associated with typosquatting and thus naturally attracts more traffic than domains that are just soundsquatting. Apart from requests towards the main page of each website, we recorded many requests towards subdomains within each domain. For instance, `jimdo.com` is a web application where users can create their own websites and host them on subdomains of the main domain. In our `jimdoe.com` logs, we found requests towards 176 subdomains connected to personal sites, such as `awesomegrizzlybears.jimdoe.com`, `karatedojo-oppeln.jimdoe.com` and `armaniwoe.jimdoe.com`. These were all valid subdomains within the main `jimdo.com` domain and thus their visits show that, even though people can accurately type relatively long and obscure subdomains, they can still confuse homophone words.

By geolocating the IP addresses of all requests we discovered that, while there were 42 countries involved in the crawling of our domains, requests from human visitors originated from 123 different countries, demonstrating that users from all countries are prone to homophone confusion and thus vulnerable to soundsquatting attacks.

In general, each soundsquatting domain received between 2 and 283 human requests per month. While these numbers are not incredibly large, and probably smaller than popular typosquatting domains, it is important to remind the reader that soundsquatting and typosquatting are not competing techniques, but rather complementing ones in the arsenal of domain squatters. Moreover, since we are the first ones to study soundsquatting, we registered domains with homophone replacements ranging from more likely, to less likely. Careful attackers could target domains in a better way, and thus acquire more users for a smaller cost.

Finally, it is worthwhile mentioning that we received a significant number of emails sent to our soundsquatting domains. Among others, we received social networking invitations, notifications of the shipment of various products, account-creation emails with credentials, and bills of mobile telephony. In all cases, it was evident that the emails were meant to be sent to accounts belonging to the legitimate domains which we targeted, but were sent to us due to homophone confusion. The receipt of these emails further demonstrates that businesses and users are indeed vulnerable to soundsquatting attacks.

5 Sound-dependent users

In previous sections, we described the soundsquatting attack and we investigated the existing soundsquatting domains as well as the users' susceptibility to homophone confusion. In this section, we describe a variation of the attack that is geared towards people that rely on sound.

According to the Word Health Organization, there are currently 285 million people that are visually impaired, of which 39 million are blind [2]. People that

are severely visually impaired, cannot properly interact with a computer without the use of assistive technologies. The two most popular assistive technologies for the visually impaired are Braille displays and screen readers [9]. In both cases, the assistive technology converts content that would be consumed by sight, into content that can be consumed by touch or sound. If one thinks of the definition of homophones and their relation to soundsquatting, a new attack becomes clear.

A user that depends on a screen reader in order to consume content in emails, web pages, messages in social networks or instant messaging applications, is vulnerable to links pointing to soundsquatting domains. A soundsquatting domain will be "read" near-identical with the targeted authoritative domain and thus the visually impaired user has no reason not to click on the offered link. While Braille displays are not vulnerable to this attack, the fact that about 90% of the visually impaired people live in developing countries combined with the high cost of Braille displays, suggest that, due to limited resources and possible portability issues, screen readers are used much more than Braille devices. Moreover, apart from visually-impaired users, hundreds of thousands of smartphone users utilize personal assistant software, like Apple's Siri, with text-to-speech capabilities when involved in another activity that makes it hard to operate their smartphones, like driving or running.

To test our theory, we sent to a web-mail account an email with two links, one pointing to `youtube.com` and one pointing to `yewtube.com`. We had our email read to us by five popular free screen readers, i.e., by the built-in screen reader of Windows XP, Windows 7 and Mac OS X, the Linux-based, open-source ORCA [23], and the Thunder screen reader [26]. We also sent a text-message with the same information to an Android smartphone with Skyvi [5], a popular Siri-like application installed by more than 260,000 users.

In all six cases, the two links sounded identical to each other which means that a sound-dependent person would have no means of separating a legitimate link from a malicious one. To further exacerbate the issue, this type of attack can also work with pseudohomophones, i.e., combinations of characters that are not real dictionary words but are purposefully constructed to sound like real words, such as {*joke, joak*} [24]. Thus, pseudo-soundsquatting domains can be crafted even for target domains that contain no homophones, such as `phacebook.com` and `phaceboocc.com`.

Due to the potentially large number of such domain variations and the specificity of this attack, we reason that the responsibility of protecting a sound-dependent user should be on the text-to-speech software.One way of protecting against this threat is for the software to switch to a "spelling mode" whenever a link is encountered, so that the user will realize that the link is not what it sounds like and avoid visiting the malicious website.

6 Limitations & Future work

In Section 2.3 we described the workings of AutoSS, a tool that automatically generates soundsquatting domains. While we accounted for many corner cases

when attempting to identify the words comprising a domain name, there is, unfortunately, still room for false positives, i.e., generated domains that do not conform to our definition of soundsquatting and the intuition behind it. For instance, there are many domains in the Alexa top 10,000 which do not include English words, like `laredoute.fr`, a French e-shop. AutoSS uses an English dictionary and thus will identify the words "lare", "do" and "ute" inside the domain name. The "Accidental Words Removal" module of AutoSS, will successfully combine these words to "laredoute" and thus, the words will then be used as keys in our homophone replacement database resulting to improbable domains, such as `laredewute.fr`. Prior systems proposed to automatically generate typosquatting domains do not suffer from such problems, since they operate at a character level [21,27] whereas soundsquatting operates at a word level.

To estimate the number of false positives, we randomly sampled 5% (424) of the generated soundsquatting domains and manually examined each homophone replacement, classifying each domain as a true or false positive. At the end of this process we identified 80 false positives out of the 424 investigated domains (18.9% with a margin of sampling error ±4.75%). While the number of false positives is not negligible, the main purpose of our work was to investigate a previously unreported domain-squatting technique and to evaluate its practicality and adoption on the web which we believe that we did.

The lack of punctuation in a domain name makes identifying its language, a challenging task. One way around this problem would be to actually inspect the main page of the site, characterize the language of that, and assume that the domain name contains words of the same language. We leave the exploration of this and other techniques of reducing false positives, for future work.

7 Related work

To the best of our knowledge, this paper is the first one that uncovers the use of homophones as a way of performing domain squatting, and systematically studies the adoption of the attack, as well as the user's susceptibility towards it.

Domain squatting was the first type of cybersquatting involving the registration of domains that were trademarks belonging to other persons and companies, before the latter had a chance to register [6,8,13]. Domain squatting later evolved into *typosquatting* [8,21,27], i.e., the act of registering domains that are mistypes of popular authoritative domains which can be traced back to 1999, through the Anticybersquatting Consumer Protection Act (ACPA) which already mentions URLs that are "sufficiently similar to a trademark of a person or entity."[3]

Apart from typosquatting, there also exist other, less popular, types of domain squatting, such as domains that abuse the visual similarity of characters in different character sets [11,16], and domains that capture the traffic originating from erroneous bit-flips in user devices [7,22].

8 Conclusion

In this paper we uncovered a new type of domain squatting based on the sound similarity of words, rather than typographical mistakes. We named the attack "soundsquatting", described a system for automatically generating soundsquatting domains, and showed that attackers are already familiar with the concepts behind soundsquatting, abusing them in ways similar to known types of domain squatting. By registering our own soundsquatting domains, we showed that it is possible for well-selected soundsquatting domains to attract hundreds of human visitors every month. We also briefly examined the relationship between text-to-speech software and soundsquatting, and showed that attackers could abuse the former to trick sound-dependent users into visiting malicious soundsquatting and pseudo-soundsquatting domains. Overall, our findings verify the practicality of soundsquatting and show that homophone confusion should be accounted for, by website owners, registrars, and cybersquatting countermeasures.

Acknowledgments: We want to thank the anonymous reviewers for the valuable comments. This research was performed with the financial support of the Prevention against Crime Programme of the European Union (B-CCENTRE), the Research Fund KU Leuven, and the EU FP7 projects NESSoS and STREWS.

References

1. IP Addresses of Search Engine Spiders. `http://iplists.com/`.
2. WHO — Visual impairment and blindness. `http://www.who.int/mediacentre/factsheets/fs282/en/`.
3. Anticybersquatting Consumer Protection Act (ACPA). `http://www.patents.com/acpa.htm`, November 1999.
4. A. Banerjee, D. Barman, M. Faloutsos, and L. N. Bhuyan. Cyber-fraud is one typo away. In *Proceedings of IEEE INFOCOM*, 2008.
5. BlueTornado. Skyvi (Siri for Android). `http://www.skyviapp.com`.
6. S. E. Coull, A. M. White, T.-f. Yen, F. Monrose, and M. K. Reiter. Understanding domain registration abuses. In *IFIP SEC'10*, 2010.
7. A. Dinaburg. Bitsquatting: DNS Hijacking without Exploitation. In *Proceedings of BlackHat Security*, July 2011.
8. B. Edelman. Large-scale registration of domains with typographical errors, 2003.
9. Even Grounds - How Do Blind People Use The Computer. `http://www.evengrounds.com/blog/how-do-blind-people-use-the-computer`.
10. R. Ferguson. Tvviter Typosquatting Phishing Site. `http://countermeasures.trendmicro.eu/tvviter-typosquatting-phishing-site/`.
11. E. Gabrilovich and A. Gontmakher. The homograph attack. *Communucations of the ACM*, 45(2):128, Feb. 2002.
12. G. Gee and P. Kim. Doppelganger Domains. `http://www.wired.com/images_blogs/threatlevel/2011/09/Doppelganger.Domains.pdf`.
13. J. Golinveaux. What's in a domain name: Is cybersquatting trademark dilution? In *University of San Francisco Law Review 33 U.S.F. L. Rev. (1998-1999)*.
14. A. Herzberg and H. Shulman. Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org. In *CNS '13*, pages 224–232. IEEE, 2013.

15. A. Hidayat. PhantomJS: Headless WebKit with JavaScript API.
16. T. Holgers, D. E. Watson, and S. D. Gribble. Cutting through the confusion: a measurement study of homograph attacks. In *Proceedings of USENIX ATC*, 2006.
17. M. Jakobsson, P. Finn, and N. Johnson. Why and How to Perform Fraud Experiments. *Security & Privacy, IEEE*, 6(2):66–68, March-April 2008.
18. M. Jakobsson and J. Ratkiewicz. Designing ethical phishing experiments: a study of (ROT13) rOnl query features. In *WWW '06*, 2006.
19. D. Kesmodel. *The Domain Game: How People Get Rich from Internet Domain Names*. Xlibris Corporation, 2008.
20. R. McMahon. BIND 8.2 NXT Remote Buffer Overflow Exploit, 2000.
21. T. Moore and B. Edelman. Measuring the perpetrators and funders of typosquatting. In *Financial Cryptography and Data Security*, pages 175–191, 2010.
22. N. Nikiforakis, S. V. Acker, W. Meert, L. Desmet, F. Piessens, and W. Joosen. Bitsquatting: Exploiting bit-flips for fun, or profit? In *WWW'13*, pages 989–998, 2013.
23. Orca: a free, open source, flexible, and extensible screen reader.
24. M. S. Seidenberg, A. Petersen, M. C. MacDonald, and D. C. Plaut. Pseudohomophone Effects and Models of Word Recognition. In *Journal of Experimental Psychology: Learning, Memory and Cognition*, volume 22, pages 48–62, 1996.
25. J. Stewart. DNS Cache Poisoning - The Next Generation, 2003.
26. ScreenReader.net: freedom for blind and Visually impaired people.
27. Y.-M. Wang, D. Beck, J. Wang, C. Verbowski, and B. Daniels. Strider typo-patrol: discovery and analysis of systematic typo-squatting. SRUTI'06, 2006.
28. List of dialect-independent homophones. `http://en.wiktionary.org/wiki/Appendix:List_of_dialect-independent_homophones`.

A Appendix

Ethical considerations Registering soundsquatting domains and receiving user traffic to these domains may raise ethical concerns. However, analogous to the real-world experiments conducted by Jakobsson et al. [17,18], we believe that realistic experiments are the only way to reliably estimate success rates of attacks in the real world. Moreover, we believe that our findings will help websites to protect their brands and customers.

The data that was collected for this experiment are the following: For each request to our soundsquatting domains we recorded i) its timestamp ii) the IP address of the host performing the request, iii) domain, path and GET parameters and iv) the user agent, as provided by the Apache web server. This kind of data is collected by every web server on the web in standard server logs and many web developers even share this collected information with third parties, such as Google Analytics, for the purpose of gathering usage statistics. The server logs were only accessible to the authors of this paper. Similarly, the emails were all collected in a single, password-protected email account of one of the authors. We did not attempt to extract any information from these emails, nor trace their senders. Gee and Kim performed a similar experiment in 2011, capturing emails through typosquatting domains and released statistics to the research community, as a demonstration of the dangers of typosquatting [12].